

```

////////////////////////////////////
//          Binary Logocal Computer Using ARDUINO UNO R3          //
//          ~~~~~~                                         //
//          Develoed by : Mr. Vaibhav K Sugandhi             //
//          Founder of Unique Tech Designs - Hubli         //
//          uniquetechdesigns555@gmail.com                 //
//          Contact : 74118-04972                         //
//          =====                                         //
//          This code has 3 select line based on which 8-bit data is computed //
//          Refer Opcode assignment to understand combination of required operation //
////////////////////////////////////

/*////////////////////////////////////*/
/* Code starts from here*/
const int Time = 100;
int q = 0, Opcode = 0, i = 0, cin = 0, c1 = 0, c2 = 0, c3 = 0, c4 = 0;
int B[15];
short int Outputs[] = {A0, A1, A2, A3, A4, A5, 0, 1};
void setup()
{
  for (i = 0; i < 9; i++)
  {
    pinMode(Outputs[i], OUTPUT);
  }
  for (i = 0; i < 9; i++)
  {
    digitalWrite(Outputs[i], HIGH);
    delay(Time);
  }
  for (i = 0; i < 9; i++)
  {
    digitalWrite(Outputs[i], LOW);
    delay(Time);
  }
  for (i = 2; i < 13; i++)
  {
    pinMode(i, INPUT);
    digitalWrite(i, LOW);
  }
  pinMode(13, OUTPUT);
}
void loop()
{
  digitalWrite(13, HIGH);
  delay(100);
  digitalWrite(13, LOW);
  if (digitalRead(2) == LOW && digitalRead(3) == LOW && digitalRead(4) == LOW)
    Opcode = 000; // Binary to Gray Code conversion
  else if (digitalRead(2) == LOW && digitalRead(3) == LOW && digitalRead(4) == HIGH)
    Opcode = 001; // Gray to Binary Code conversion
  else if (digitalRead(2) == LOW && digitalRead(3) == HIGH && digitalRead(4) == LOW)
    Opcode = 010; //BCD to Ex-3 Code conversion
  else if (digitalRead(2) == LOW && digitalRead(3) == HIGH && digitalRead(4) == HIGH)
    Opcode = 011; //Ex-3 to BCD Code conversion
  else if (digitalRead(2) == HIGH && digitalRead(3) == LOW && digitalRead(4) == LOW)
    Opcode = 100; //Binary 4-Bit parellel Adder
  else if (digitalRead(2) == HIGH && digitalRead(3) == LOW && digitalRead(4) == HIGH)
    Opcode = 101; //Binary 4-Bit Subtractor
  else if (digitalRead(2) == HIGH && digitalRead(3) == HIGH && digitalRead(4) == LOW)
    Opcode = 110; //4-bit counters(UP/DOWN)
  else if (digitalRead(2) == HIGH && digitalRead(3) == HIGH && digitalRead(4) == HIGH)
    Opcode = 111; //4 - bit Magnitude comparator
  else
  {
    // Do nothing
  }
  switch (Opcode)
  {
    case (000): // Binary to Gray Code conversion
      {
        for(i = 5; i<13 ; i++)
        {
          B[i] = digitalRead(i);
        }
        digitalWrite(1,B[12]);
        digitalWrite(0,(B[11]^B[12]));
        digitalWrite(A5,(B[10]^B[11]));
        digitalWrite(A4,(B[9]^B[10]));
        digitalWrite(A3,(B[8]^B[9]));
        digitalWrite(A2,(B[7]^B[8]));
        digitalWrite(A1,(B[6]^B[7]));
        digitalWrite(A0,(B[5]^B[6]));
        break;
      }
    case (001): // Gray to Binary Code conversion
      {
        for(i = 5; i<13 ; i++)
        {
          B[i] = digitalRead(i);
        }
        digitalWrite(1,B[12]);
        digitalWrite(0,(B[12]^B[11]));
        digitalWrite(A5,((B[12]^B[11])^B[10]));
        digitalWrite(A4,((B[12]^B[11])^B[10])^B[9]));
      }
  }
}

```

```

digitalWrite(A3,((( (B[12]^B[11]) ^ B[10]) ^ B[9]) ^ B[8]));
digitalWrite(A2,((( (B[12]^B[11]) ^ B[10]) ^ B[9]) ^ B[8]) ^ B[7]));
digitalWrite(A1,(((( (B[12]^B[11]) ^ B[10]) ^ B[9]) ^ B[8]) ^ B[7]) ^ B[6]));
digitalWrite(A0,(((( (B[12]^B[11]) ^ B[10]) ^ B[9]) ^ B[8]) ^ B[7]) ^ B[6]) ^ B[5]));
break;
}
case (010)://BCD to Ex-3 Code conversion
{
for(i = 5; i<13 ; i++)
{
B[i] = digitalRead(i);
}
if(B[5] == LOW & B[6] == LOW & B[7] == LOW & B[8] == LOW & B[9] == LOW & B[10] == LOW & B[11] == LOW & B[12] == LOW)
{
for(i = 2; i<8; i++)
{
digitalWrite(Outputs[i],LOW);
}
digitalWrite(A1,HIGH);
digitalWrite(A0,HIGH);
}
else if(B[5] == HIGH & B[6] == LOW & B[7] == LOW & B[8] == LOW & B[9] == LOW & B[10] == LOW & B[11] == LOW & B[12] == LOW)
{
for(i = 3; i<8; i++)
{
digitalWrite(Outputs[i],LOW);
}
digitalWrite(A2,HIGH);
digitalWrite(A1,LOW);
digitalWrite(A0,LOW);
}
else if(B[5] == LOW & B[6] == HIGH & B[7] == LOW & B[8] == LOW & B[9] == LOW & B[10] == LOW & B[11] == LOW & B[12] == LOW)
{
for(i = 3; i<8; i++)
{
digitalWrite(Outputs[i],LOW);
}
digitalWrite(A2,HIGH);
digitalWrite(A1,LOW);
digitalWrite(A0,HIGH);
}
else if(B[5] == HIGH & B[6] == HIGH & B[7] == LOW & B[8] == LOW & B[9] == LOW & B[10] == LOW & B[11] == LOW & B[12] == LOW)
{
for(i = 3; i<8; i++)
{
digitalWrite(Outputs[i],LOW);
}
digitalWrite(A2,HIGH);
digitalWrite(A1,HIGH);
digitalWrite(A0,LOW);
}
else if(B[5] == LOW & B[6] == LOW & B[7] == HIGH & B[8] == LOW & B[9] == LOW & B[10] == LOW & B[11] == LOW & B[12] == LOW)
{
for(i = 3; i<8; i++)
{
digitalWrite(Outputs[i],LOW);
}
digitalWrite(A2,HIGH);
digitalWrite(A1,HIGH);
digitalWrite(A0,HIGH);
}
else if(B[5] == HIGH & B[6] == LOW & B[7] == HIGH & B[8] == LOW & B[9] == LOW & B[10] == LOW & B[11] == LOW & B[12] == LOW)
{
for(i = 4; i<8; i++)
{
digitalWrite(Outputs[i],LOW);
}
digitalWrite(A3,HIGH);
digitalWrite(A2,LOW);
digitalWrite(A1,LOW);
digitalWrite(A0,LOW);
}
else if(B[5] == LOW & B[6] == HIGH & B[7] == HIGH & B[8] == LOW & B[9] == LOW & B[10] == LOW & B[11] == LOW & B[12] == LOW)
{
for(i = 4; i<8; i++)
{
digitalWrite(Outputs[i],LOW);
}
digitalWrite(A3,HIGH);
digitalWrite(A2,LOW);
digitalWrite(A1,LOW);
digitalWrite(A0,HIGH);
}
else if(B[5] == HIGH & B[6] == HIGH & B[7] == HIGH & B[8] == LOW & B[9] == LOW & B[10] == LOW & B[11] == LOW & B[12] == LOW)
{
for(i = 4; i<8; i++)
{
digitalWrite(Outputs[i],LOW);
}
digitalWrite(A3,HIGH);
digitalWrite(A2,LOW);
digitalWrite(A1,HIGH);
digitalWrite(A0,LOW);
}
}

```

```

else if(B[5] == LOW & B[6] == LOW & B[7] == LOW & B[8] == HIGH & B[9] == LOW & B[10] == LOW & B[11] == LOW & B[12] == LOW)
{
    for(i = 4; i<8; i++)
    {
        digitalWrite(Outputs[i],LOW);
    }
    digitalWrite(A3,HIGH);
    digitalWrite(A2,LOW);
    digitalWrite(A1,HIGH);
    digitalWrite(A0,HIGH);
}
else if(B[5] == HIGH & B[6] == LOW & B[7] == LOW & B[8] == HIGH & B[9] == LOW & B[10] == LOW & B[11] == LOW & B[12] == LOW)
{
    for(i = 4; i<8; i++)
    {
        digitalWrite(Outputs[i],LOW);
    }
    digitalWrite(A3,HIGH);
    digitalWrite(A2,HIGH);
    digitalWrite(A1,LOW);
    digitalWrite(A0,LOW);
}
else
{
    for(i = 0; i <8; i++)
    {
        digitalWrite(Outputs[i],HIGH);
        delay(100);
        digitalWrite(Outputs[i],LOW);
        delay(100);
    }
}
break;
}
case (011)://Ex-3 to BCD Code conversion
{
    for(i = 5; i<13 ; i++)
    {
        B[i] = digitalRead(i);
    }
    if(B[5] == HIGH & B[6] == HIGH & B[7] == LOW & B[8] == LOW & B[9] == LOW & B[10] == LOW & B[11] == LOW & B[12] == LOW)
    {
        for(i = 0; i<8; i++)
        {
            digitalWrite(Outputs[i],LOW);
        }
    }
    else if(B[5] == LOW & B[6] == LOW & B[7] == HIGH & B[8] == LOW & B[9] == LOW & B[10] == LOW & B[11] == LOW & B[12] == LOW)
    {
        for(i = 1; i<8; i++)
        {
            digitalWrite(Outputs[i],LOW);
        }
        digitalWrite(A0,HIGH);
    }
    else if(B[5] == HIGH & B[6] == LOW & B[7] == HIGH & B[8] == LOW & B[9] == LOW & B[10] == LOW & B[11] == LOW & B[12] == LOW)
    {
        for(i = 2; i<8; i++)
        {
            digitalWrite(Outputs[i],LOW);
        }
        digitalWrite(A1,HIGH);
        digitalWrite(A0,LOW);
    }
    else if(B[5] == LOW & B[6] == HIGH & B[7] == HIGH & B[8] == LOW & B[9] == LOW & B[10] == LOW & B[11] == LOW & B[12] == LOW)
    {
        for(i = 2; i<8; i++)
        {
            digitalWrite(Outputs[i],LOW);
        }
        digitalWrite(A1,HIGH);
        digitalWrite(A0,HIGH);
    }
    else if(B[5] == HIGH & B[6] == HIGH & B[7] == HIGH & B[8] == LOW & B[9] == LOW & B[10] == LOW & B[11] == LOW & B[12] == LOW)
    {
        for(i = 3; i<8; i++)
        {
            digitalWrite(Outputs[i],LOW);
        }
        digitalWrite(A2,HIGH);
        digitalWrite(A1,LOW);
        digitalWrite(A0,LOW);
    }
    else if(B[5] == LOW & B[6] == LOW & B[7] == LOW & B[8] == HIGH & B[9] == LOW & B[10] == LOW & B[11] == LOW & B[12] == LOW)
    {
        for(i = 3; i<8; i++)
        {
            digitalWrite(Outputs[i],LOW);
        }
        digitalWrite(A2,HIGH);
        digitalWrite(A1,LOW);
        digitalWrite(A0,HIGH);
    }
    else if(B[5] == HIGH & B[6] == LOW & B[7] == LOW & B[8] == HIGH & B[9] == LOW & B[10] == LOW & B[11] == LOW & B[12] == LOW)

```

```

{
  for(i = 3; i<8; i++)
  {
    digitalWrite(Outputs[i],LOW);
  }
  digitalWrite(A2,HIGH);
  digitalWrite(A1,HIGH);
  digitalWrite(A0,LOW);
}
else if(B[5] == LOW & B[6] == HIGH & B[7] == LOW & B[8] == HIGH & B[9] == LOW & B[10] == LOW & B[11] == LOW & B[12] == LOW)
{
  for(i = 3; i<8; i++)
  {
    digitalWrite(Outputs[i],LOW);
  }
  digitalWrite(A2,HIGH);
  digitalWrite(A1,HIGH);
  digitalWrite(A0,HIGH);
}
else if(B[5] == HIGH & B[6] == HIGH & B[7] == LOW & B[8] == HIGH & B[9] == LOW & B[10] == LOW & B[11] == LOW & B[12] == LOW)
{
  for(i = 4; i<8; i++)
  {
    digitalWrite(Outputs[i],LOW);
  }
  digitalWrite(A3,HIGH);
  digitalWrite(A2,LOW);
  digitalWrite(A1,LOW);
  digitalWrite(A0,LOW);
}
else if(B[5] == LOW & B[6] == LOW & B[7] == HIGH & B[8] == HIGH & B[9] == LOW & B[10] == LOW & B[11] == LOW & B[12] == LOW)
{
  for(i = 4; i<8; i++)
  {
    digitalWrite(Outputs[i],LOW);
  }
  digitalWrite(A3,HIGH);
  digitalWrite(A2,LOW);
  digitalWrite(A1,LOW);
  digitalWrite(A0,HIGH);
}
else
{
  for(i = 0; i <8; i++)
  {
    digitalWrite(Outputs[i],HIGH);
    delay(100);
    digitalWrite(Outputs[i],LOW);
    delay(100);
  }
}
break;
}
case (100)://Binary 4-Bit parellel Adder
{
  for(i = 5; i<13 ; i++)
  {
    B[i] = digitalRead(i);
  }
  cin = 0;
  digitalWrite(A0,((B[5] ^ B[9]) ^ cin));
  c1 = (((B[5] ^ B[9]) & cin) | (B[5] & B[9]));
  digitalWrite(A1,((B[6]^B[10]) ^ c1));
  c2 = (((B[6] ^ B[10]) & c1) | (B[6] & B[10]));
  digitalWrite(A2,((B[7]^B[11]) ^ c2));
  c3 = (((B[7] ^ B[11]) & c2) | (B[7] & B[11]));
  digitalWrite(A3,((B[8]^B[12]) ^ c3));
  c4 = (((B[8] ^ B[12]) & c3) | (B[8] & B[12]));
  digitalWrite(A4,c4);
  break;
}
case (101)://Binary 4-Bit Subtractor
{
  for(i = 5; i<13 ; i++)
  {
    B[i] = digitalRead(i);
  }
  cin = 0;
  digitalWrite(A0,((cin^B[9])^B[5]));
  c1 = (((~B[5])&(cin^B[9]))|(B[9]&cin));
  digitalWrite(A1,((c1^B[10])^B[6]));
  c2 = (((~B[6])&(c1^B[10]))|(B[10]&c1));
  digitalWrite(A2,((c2^B[11])^B[7]));
  c3 = (((~B[7])&(c2^B[11]))|(B[11]&c2));
  digitalWrite(A3,((c3^B[12])^B[8]));
  c4 = (((~B[8])&(c3^B[12]))|(B[12]&c3));
  digitalWrite(A4,c4);
  break;
}
case (110)://4-bit counters(UP/DOWN)
{
  for(i = 5; i<13 ; i++)
  {
    B[i] = digitalRead(i);
  }
}

```

```

}
if(B[5] == LOW & B[6] == LOW & B[7] == LOW & B[8] == LOW & B[9] == LOW & B[10] == LOW & B[11] == LOW & B[12] == LOW)
{
for(i = 0; i <8; i++)
{
digitalWrite(Outputs[i],HIGH);
delay(100);
digitalWrite(Outputs[i],LOW);
delay(100);
}
}
else if(B[5] == HIGH & B[6] == LOW & B[7] == LOW & B[8] == LOW & B[9] == LOW & B[10] == LOW & B[11] == LOW & B[12] == LOW)
{
for(byte j=0; j<=15; j++)
{
displayBinary(j);
delay(500);
}
}
else if(B[5] == LOW & B[6] == HIGH & B[7] == LOW & B[8] == LOW & B[9] == LOW & B[10] == LOW & B[11] == LOW & B[12] == LOW)
{
for(byte j=15; j>=0; j--)
{
displayBinary(j);
delay(500);
}
}
else
{
for(i=0;i<8;i++)
{
digitalWrite(Outputs[i],HIGH);
delay(Time);
}
for(i=0;i<8;i++)
{
digitalWrite(Outputs[i],LOW);
delay(Time);
}
}
break;
}
case (111)://4 - bit Magnitude comparator
{
for(i = 5; i<13 ; i++)
{
B[i] = digitalRead(i);
}
int B0b = (~B[9]);
int B1b = (~B[10]);
int B2b = (~B[11]);
int B3b = (~B[12]);
int m1 = ~(B[5]^B[9]);
int m2 = ~(B[6]^B[10]);
int m3 = ~(B[7]^B[11]);
int m4 = ~(B[8]^B[12]);
int a1 = (B0b&B[5]);
int a2 = (B1b&B[6]&m1);
int a3 = (B2b&B[7]&m1&m2);
int a4 = (B3b&B[8]&m1&m2&m3);
digitalWrite(1,(a1|a2|a3|a4));
if ((a1|a2|a3|a4) == HIGH || (~((a1|a2|a3|a4)|(m1&m2&m3&m4))) == HIGH)
{
digitalWrite(A3,LOW);
digitalWrite(A4,LOW);
}
else
{
digitalWrite(A3,HIGH);
digitalWrite(A4,HIGH);
}
digitalWrite(A0,(~((a1|a2|a3|a4)|(m1&m2&m3&m4))));
}
break;
} //End of switch
} //End of loop

```

// Sub-Function for Counters value display

```

void displayBinary(int num)
{
for(i=0; i<4; i++)
{
if(bitRead(num, i) == 1)
{
digitalWrite(Outputs[i],HIGH);
}
else
{
digitalWrite(Outputs[i],LOW);
}
}
}
}

```

//////////////////////////////////// THE END //////////////////////////////////////

////////////////////////////////////  
// For more information contact : Mr. Vaibhav K Sugandhi, Founder and CEO of Unique Tech Designs - Hubli //  
// 74118-04972 //  
// uniquetechdesigns5566@gmail.com //  
// FB: Unique Tech Designs - Hubli //  
// <https://www.facebook.com/UTDUBL/> //  
// Web: [utdhbl.blogspot.com](http://utdhbl.blogspot.com) //  
// Happy to build your ideas ..! //  
////////////////////////////////////